Clustering

전 재 욱

Embedded System 연구실
성균관대학교

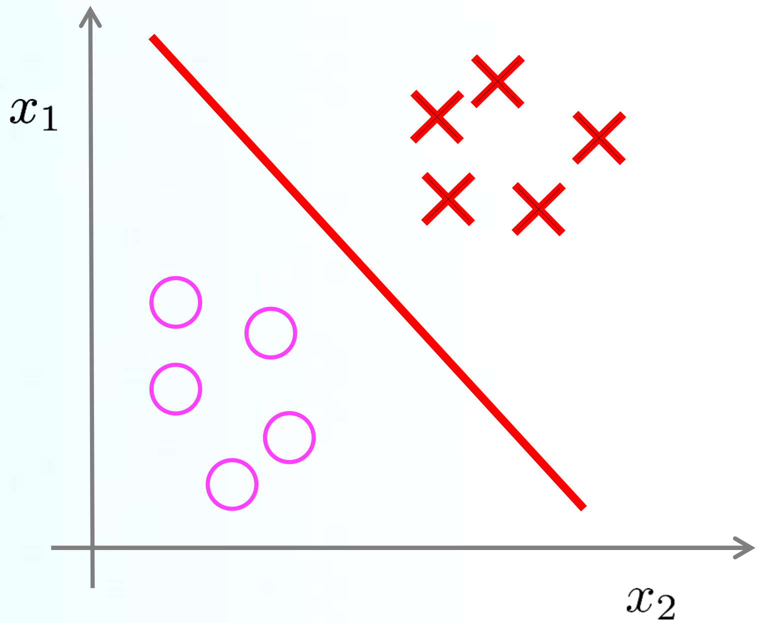
Outline

- Introduction to Unsupervised learning
- K-means algorithm
- Optimization Objective
- Random initialization
- Choosing the number of clusters

Outline

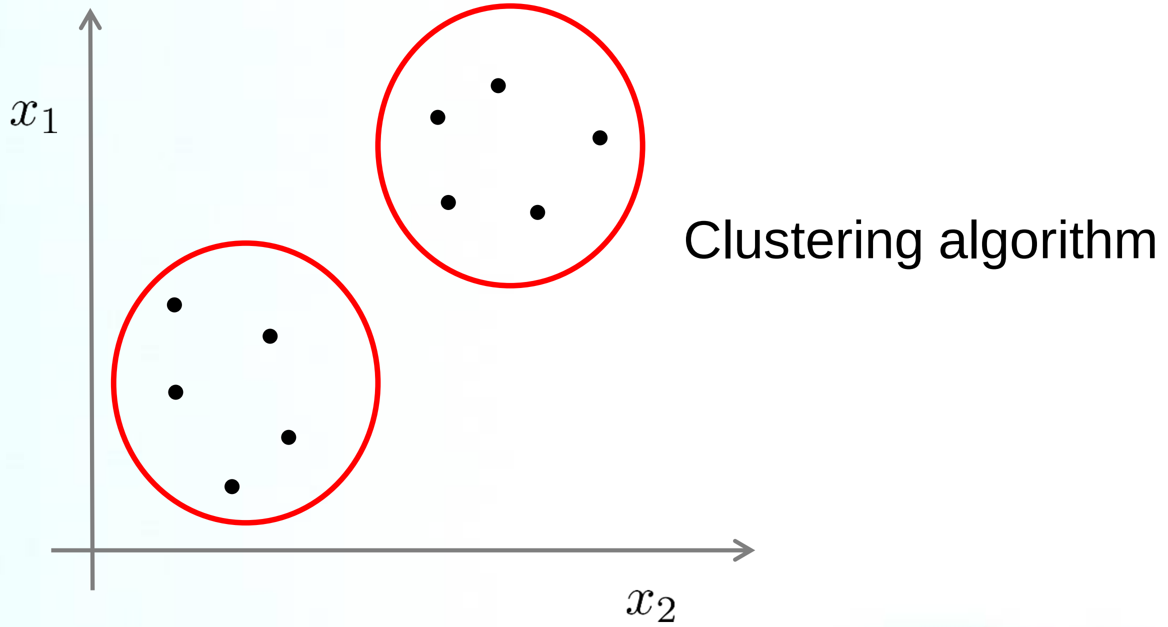
- Introduction to Unsupervised learning
- K-means algorithm
- Optimization Objective
- Random initialization
- Choosing the number of clusters

Supervised Learning



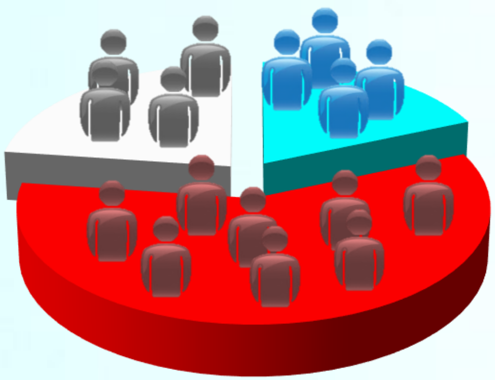
- Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)}), \}$
  - Given a set of labels, fit a hypothesis to it

Unsupervised Learning

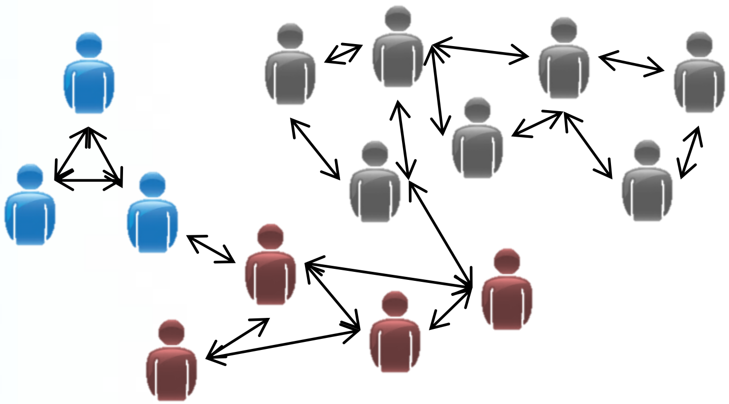


- Training set: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$
  - Try to determine structure in the data
  - Clustering algorithm groups data together based on data features

Application of Clustering



Market segmentation



Social network

Organize computing clusters

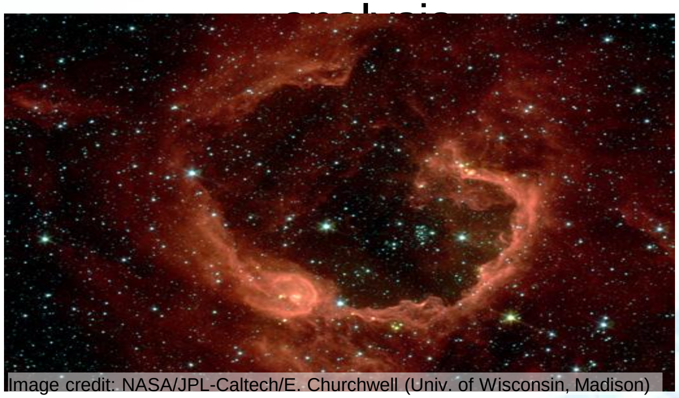


Image credit: NASA/JPL-Caltech/E. Churchwell (Univ. of Wisconsin, Madison)

Astronomical data analysis

Application of Clustering

- What is clustering good for
  - **Market segmentation**
    - Group customers into different market segments
  - **Social network analysis**
    - Facebook "smartlists"
  - **Organizing computer clusters** and data centers for network layout and location
  - **Astronomical data analysis**
    - Understanding galaxy formation

Outline

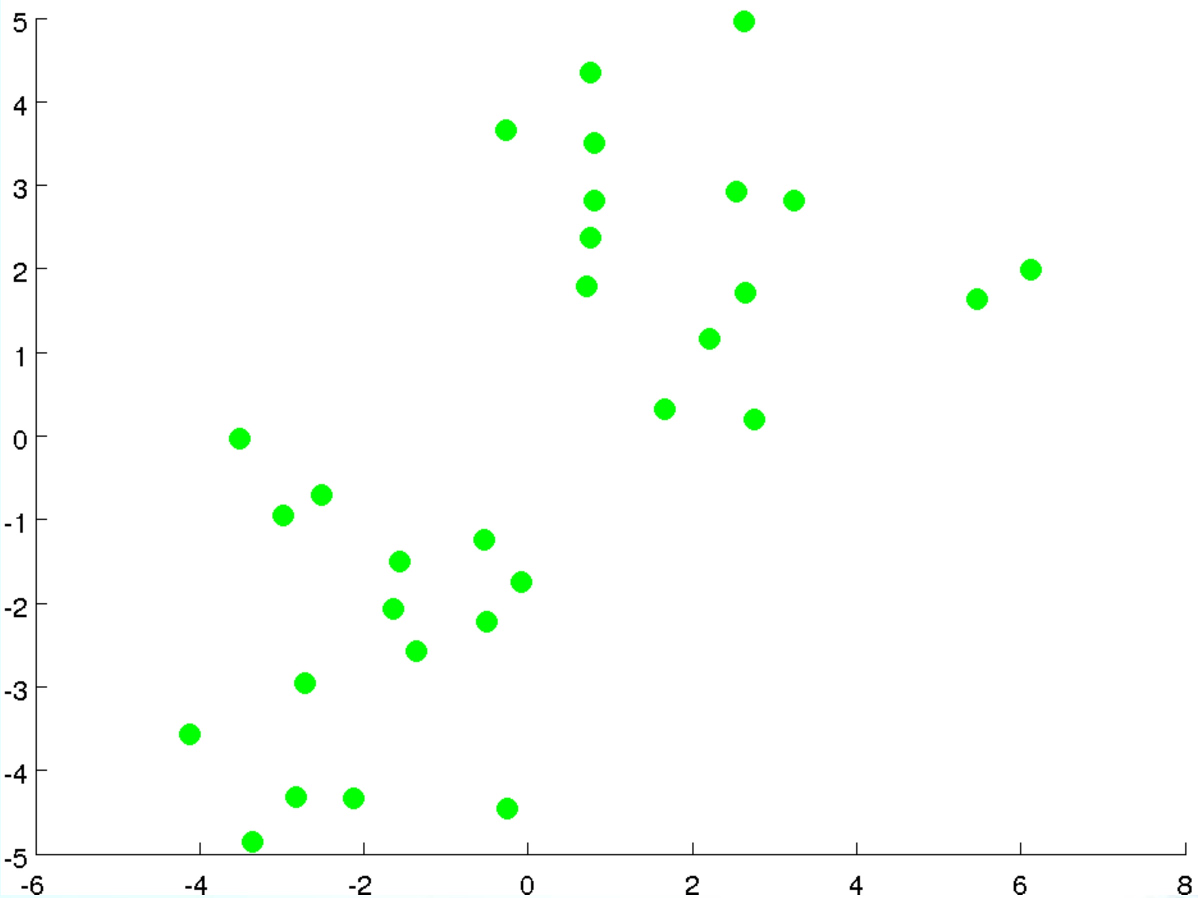
- Introduction to Unsupervised learning
- K-means algorithm
- Optimization Objective
- Random initialization
- Choosing the number of clusters

K-means Algorithm

- Want an algorithm to automatically group the data into coherent clusters

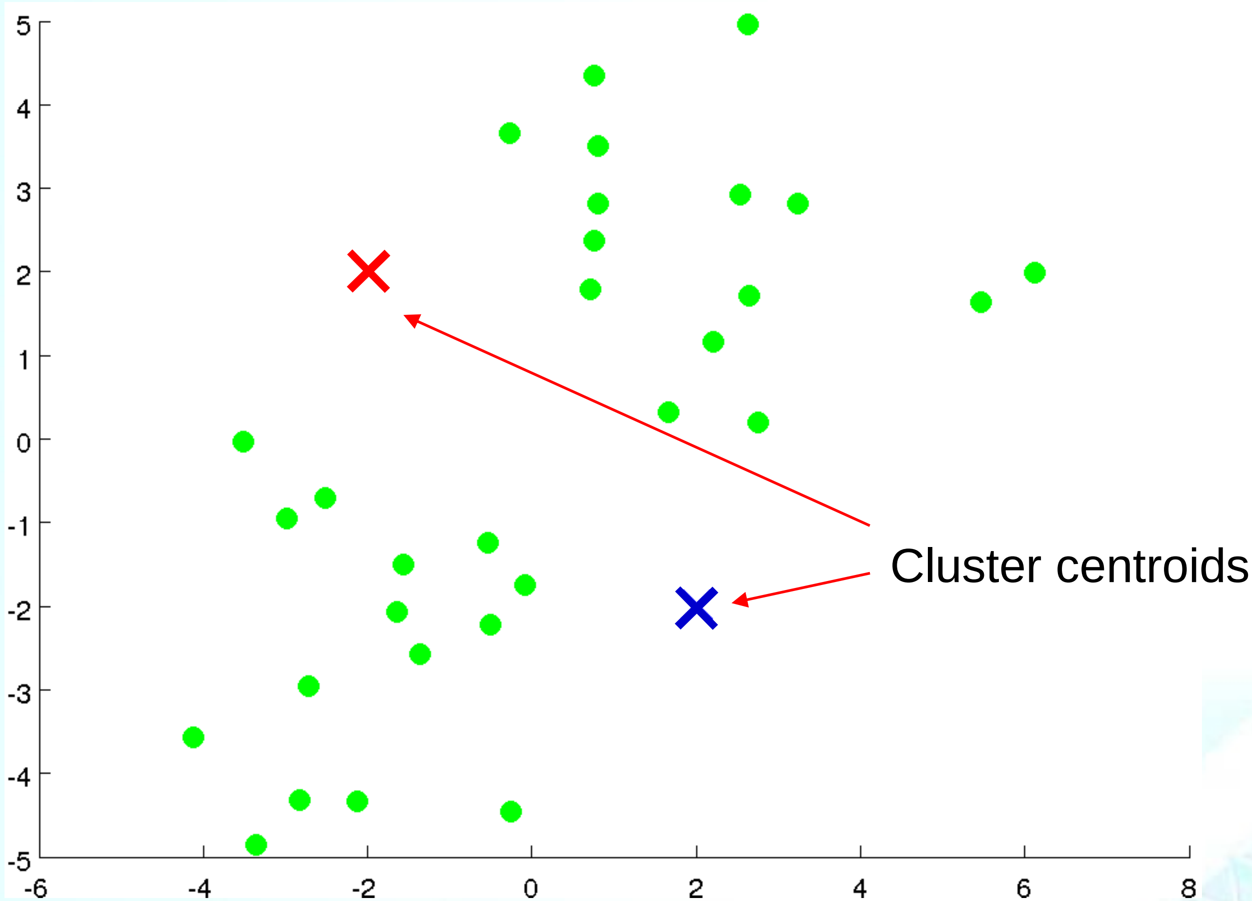
- K-means
  - The most widely used clustering algorithm

K-means Algorithm



Take unlabeled data and group into two clusters

K-means Algorithm

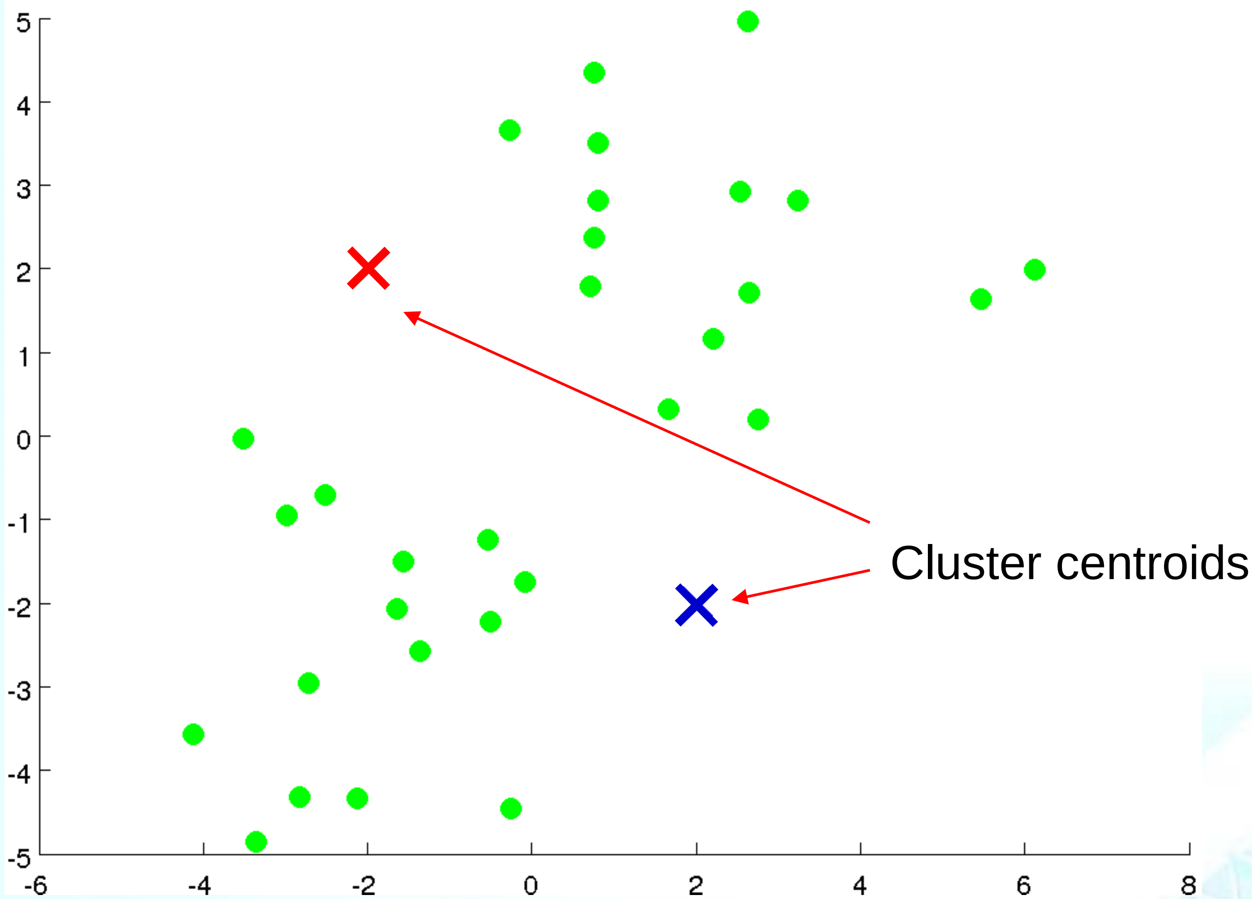


1) Randomly allocate two points as the cluster centroids

Have as many cluster centroids as clusters we want to do (*K* cluster centroids, in fact)

In this example, two clusters

K-means Algorithm

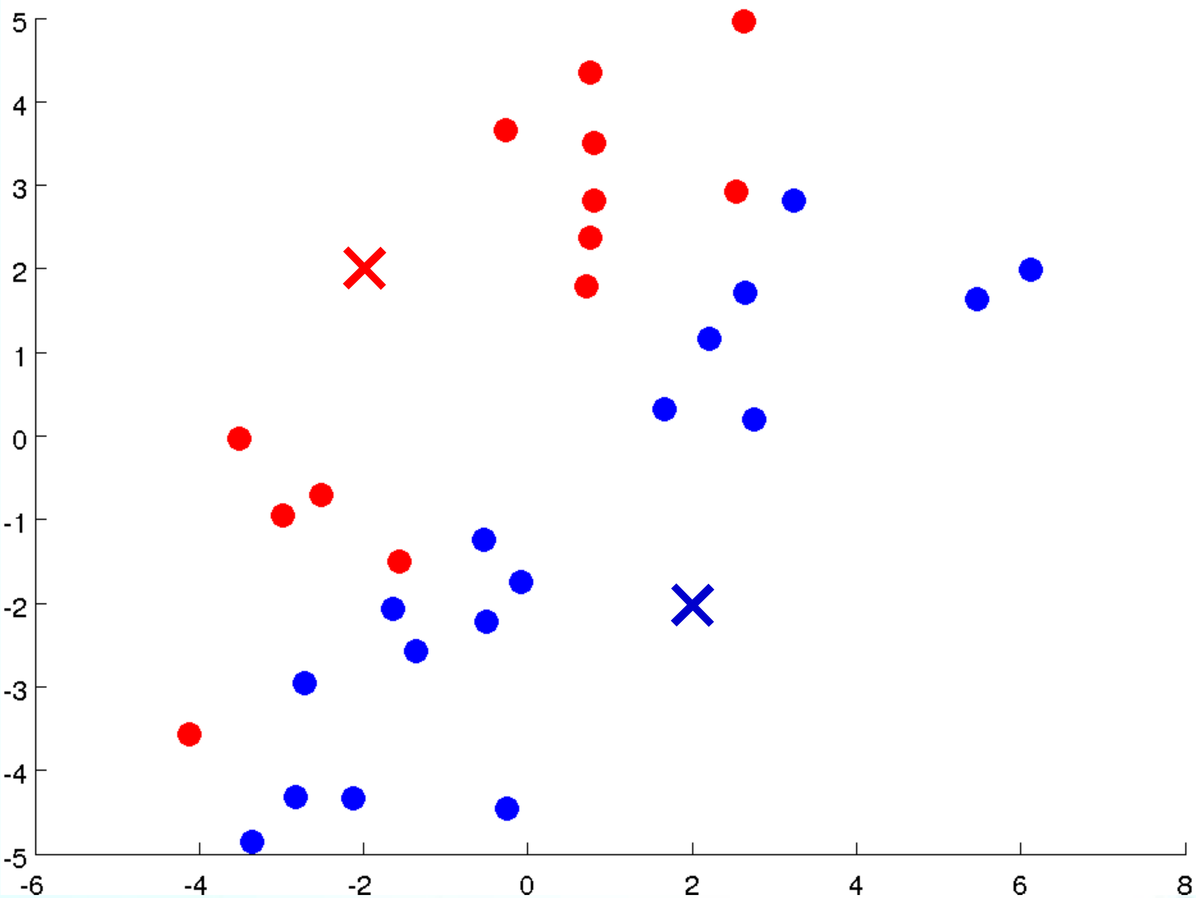


2) Cluster assignment step

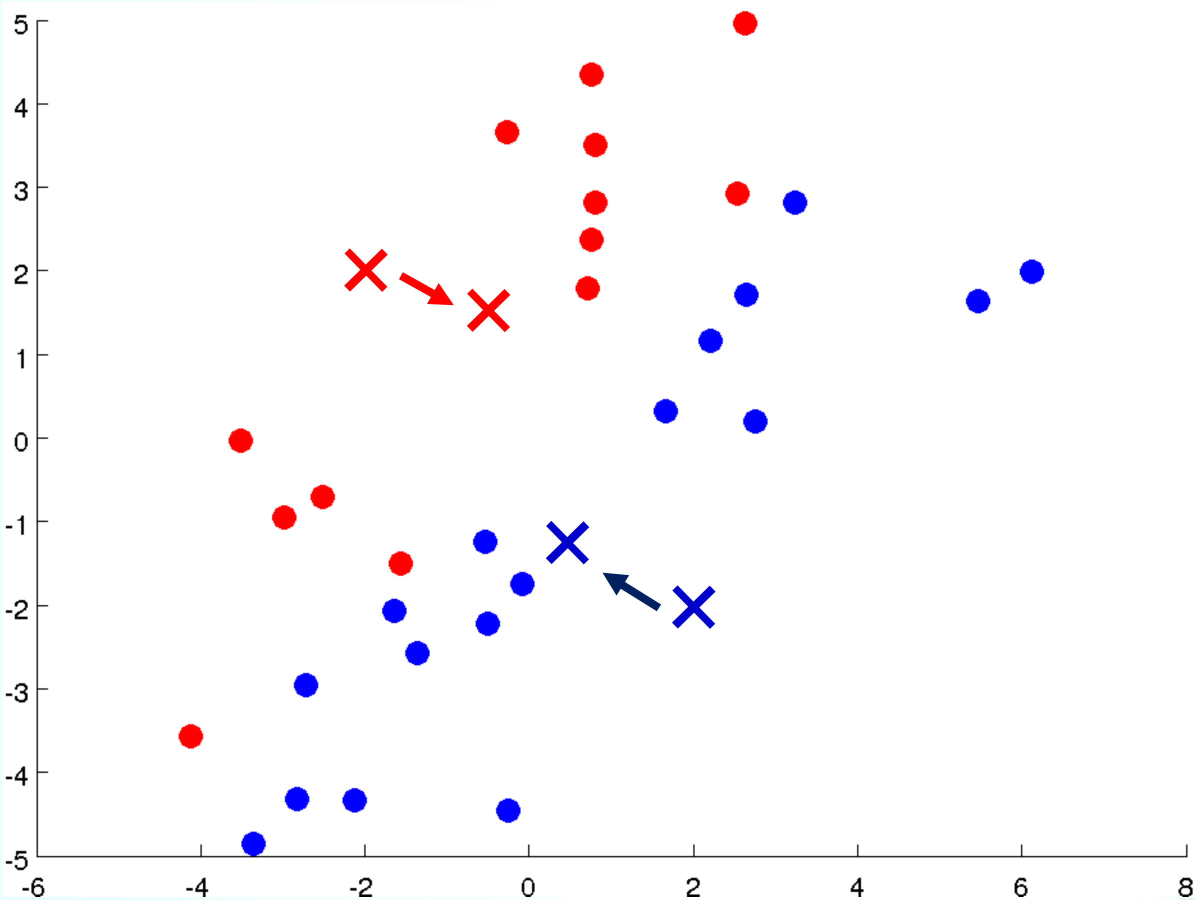
Go through each example

Check if it is closer to the red or blue centroid and assign each point to one of two clusters

K-means Algorithm



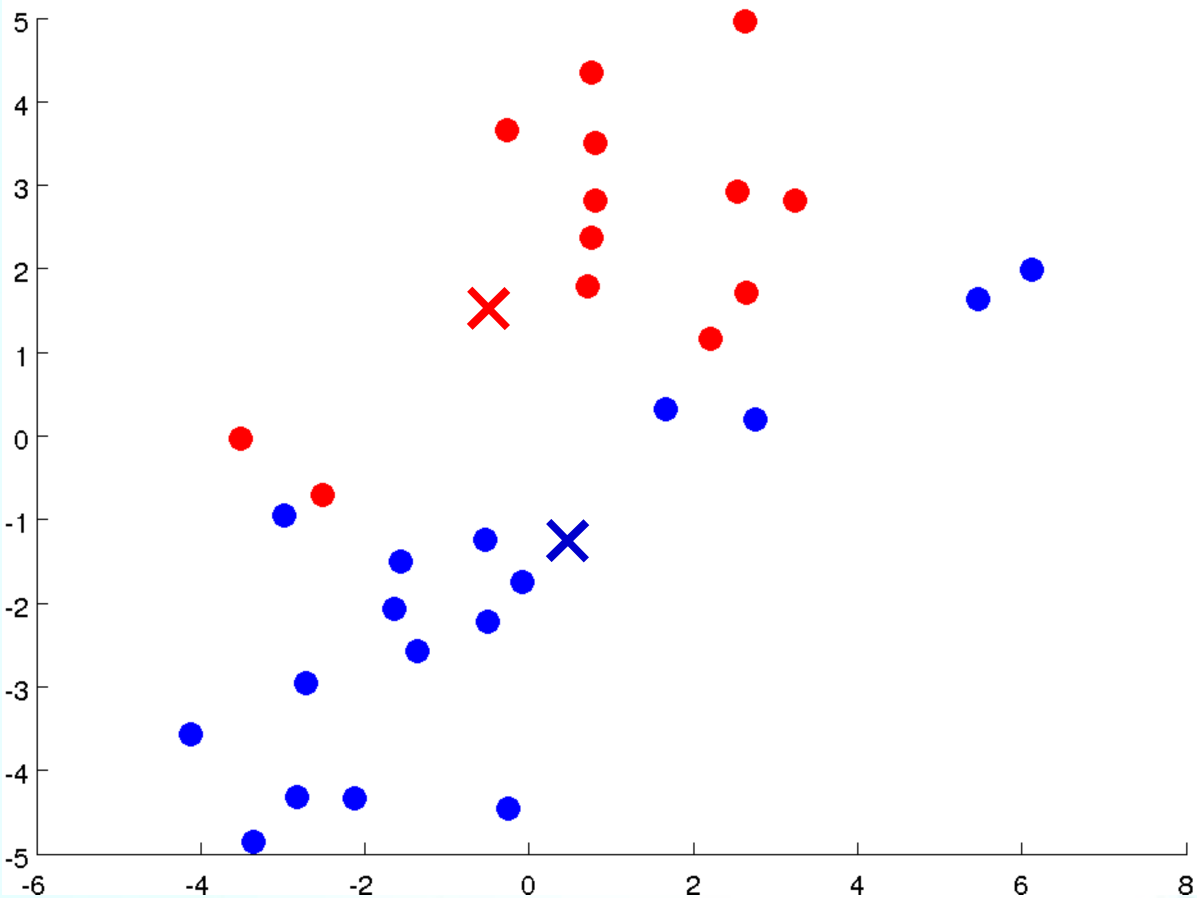
K-means Algorithm



3) Move centroid step

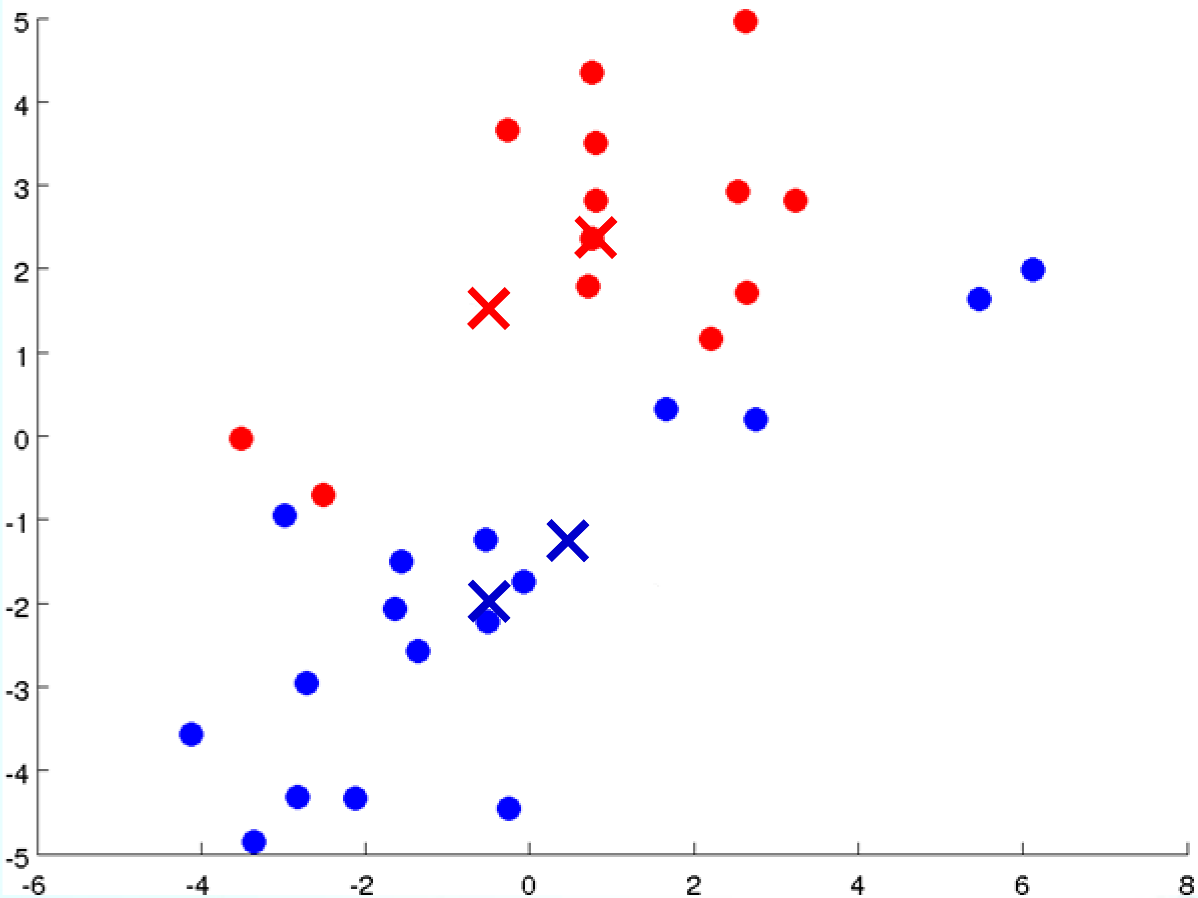
Take each centroid and move to the average of the correspondingly assigned data-points

K-means Algorithm



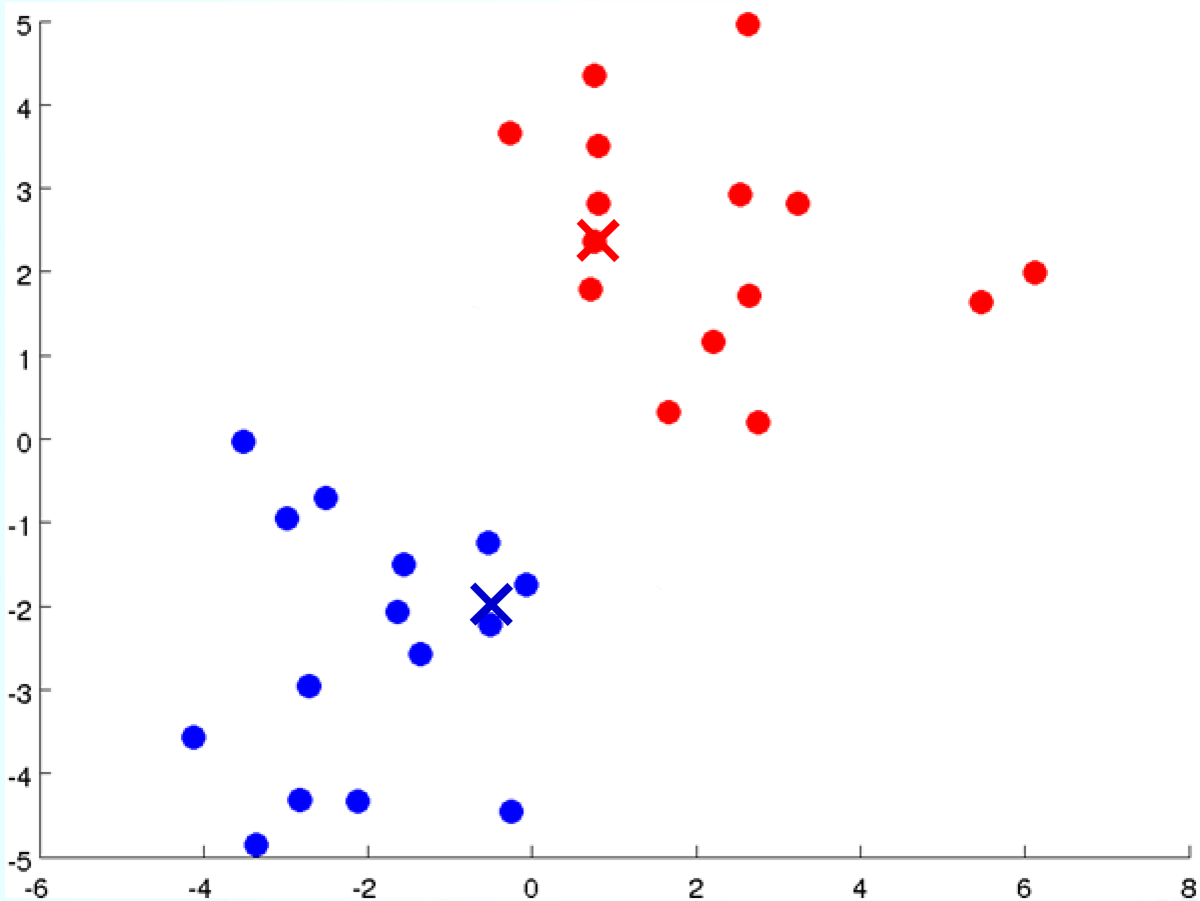
4) Repeat 2) and 3) until converging

K-means Algorithm

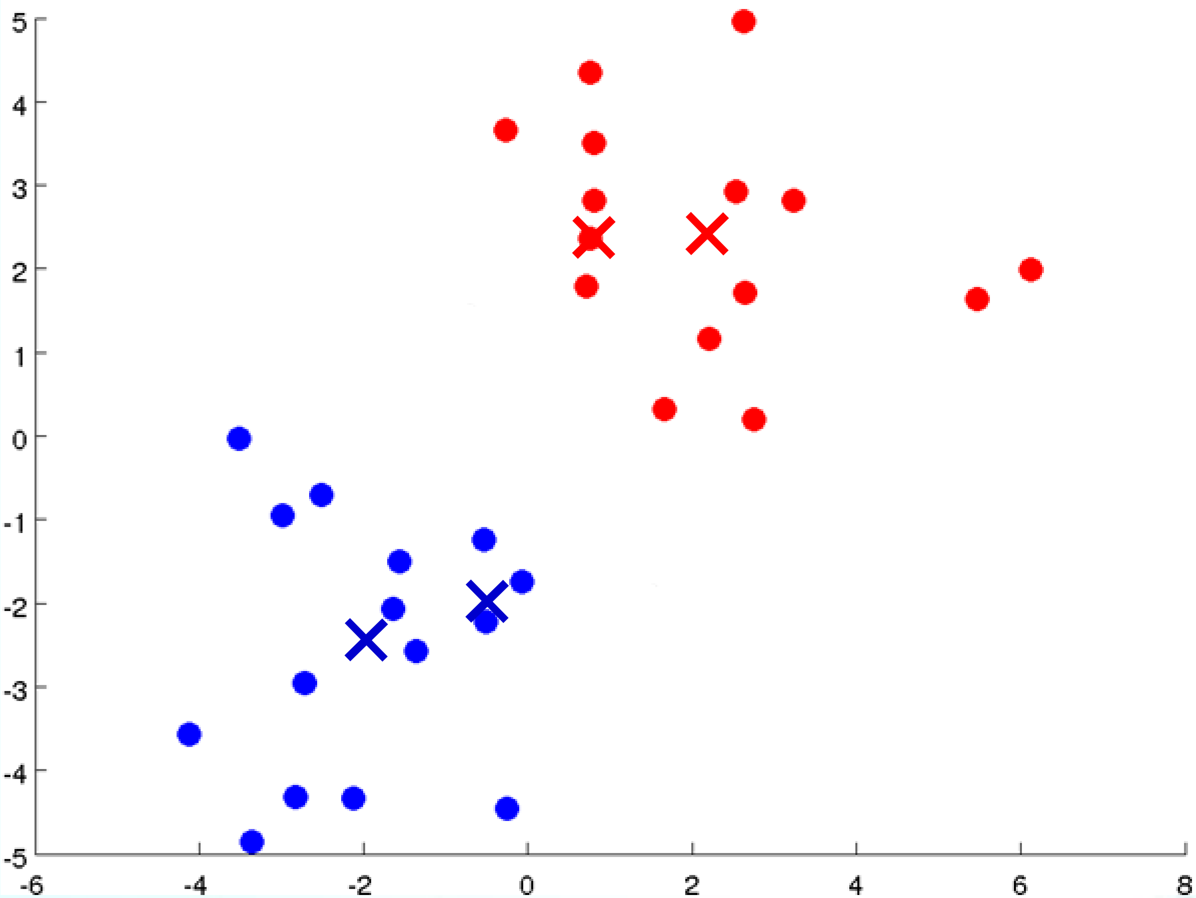


4) Repeat 2) and 3) until converging

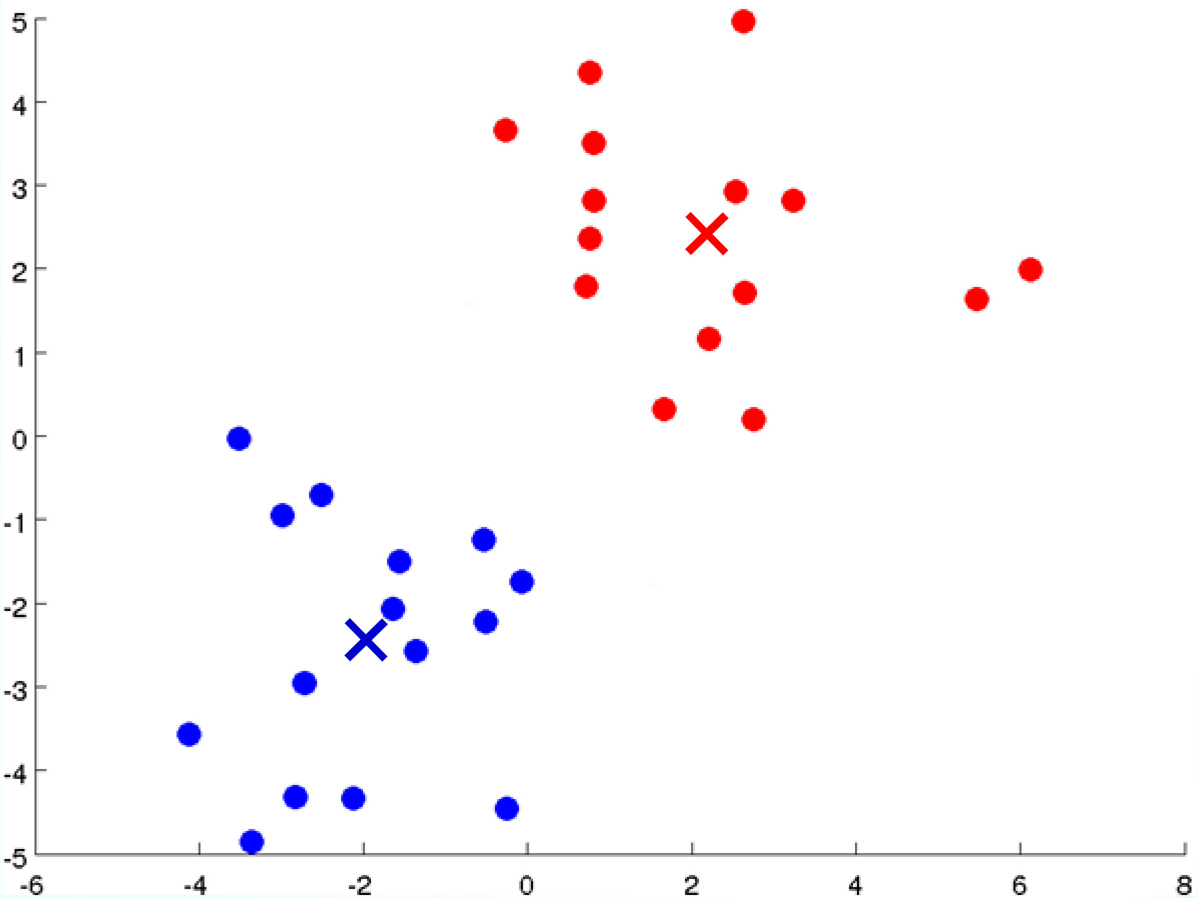
K-means Algorithm



K-means Algorithm



K-means Algorithm



K-means Algorithm

- Input:
  - K (number of clusters in the data)
  - Training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)},\}$
    - $x^{(i)} \in R^n$ (Drop $x_0 = 1$ convention)

K-means Algorithm

- Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in R^n$

Repeat {

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

for $k = 1$ to K

μ_k := average (mean) of points assigned to cluster k

}

K-means Algorithm

- Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in R^n$

Repeat {

Cluster assignment step

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

$$\min_{\boldsymbol{k}} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{\boldsymbol{k}} \right\|^2 \xrightarrow{} \boldsymbol{c}^{(i)}$$

for $k = 1$ to K

μ_k := average (mean) of points assigned to cluster k

}

Move centroid

K-means Algorithm

- Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in R^n$

Repeat {

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

$$\boxed{\min_{\boldsymbol{k}} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{\boldsymbol{k}} \right\|^{\boldsymbol{2}}} \rightarrow \boxed{\boldsymbol{c}^{(i)}}$$

Cluster assignment step

for $k = 1$ to K

μ_k := average (mean) of points assigned to cluster k

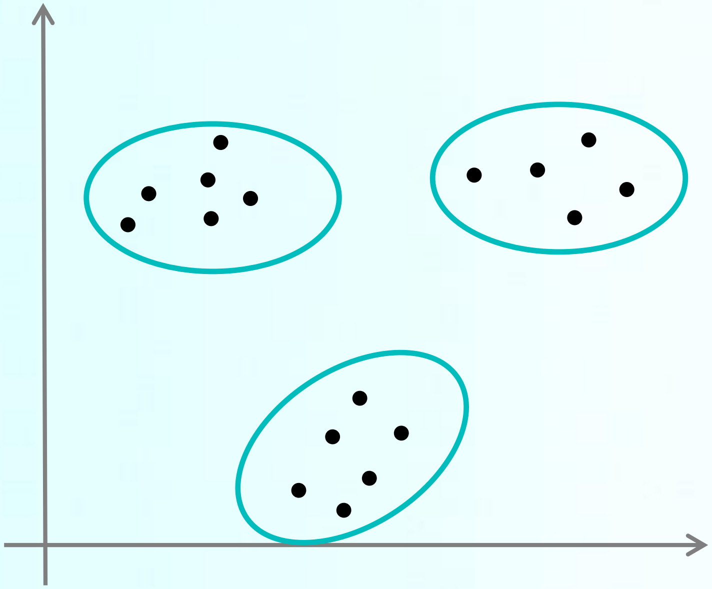
}

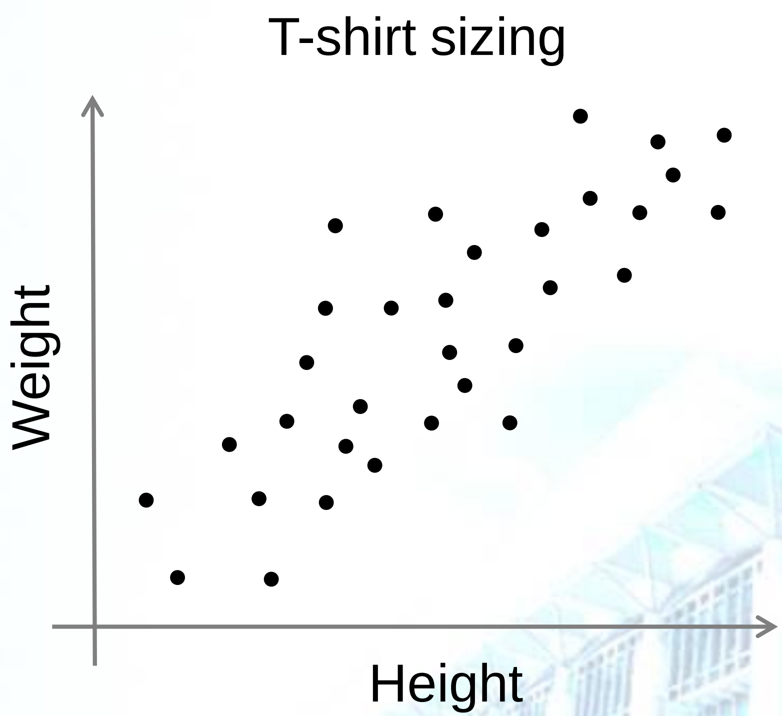
Move centroid

- Suppose that for $x^{(1)}, x^{(5)}, x^{(7)},$ and $x^{(10)}$
  - $\min_{\boldsymbol{k}} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{\boldsymbol{k}} \right\|^{\boldsymbol{2}}$ ➔ $c^{(1)} = 2,\ c^{(5)} = c^{(7)} = c^{(10)} = 2$
  - $\mu_2 = \frac{1}{4}\left(x^{(1)} + x^{(5)} + x^{(7)} + x^{(10)}\right) \in R^n$

K-means for Non-Separated Clusters

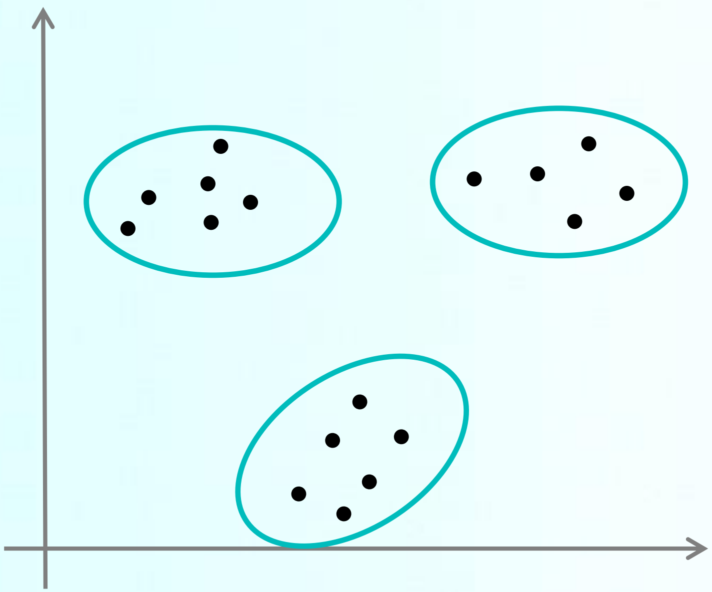
- K-means is applied to datasets where there are not well defined clusters
  - e.g. T-shirt sizing
    - Not obvious discrete groups

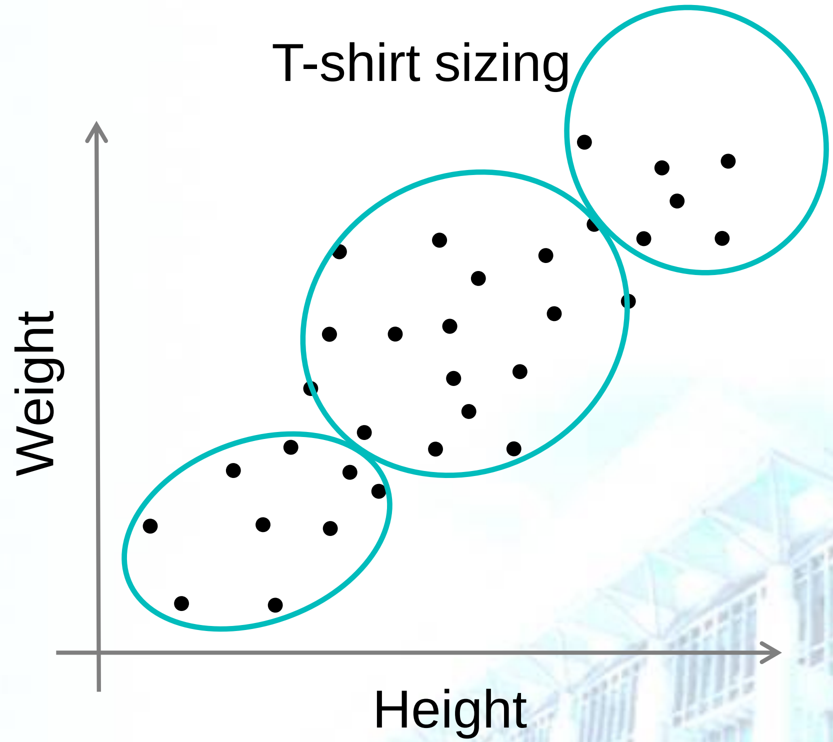




K-means for Non-Separated Clusters

- K-means is applied to datasets where there are not well defined clusters
  - e.g. T-shirt sizing
    - Not obvious discrete groups





K-means for Non-Separated Clusters

- For three sizes (S,M,L), how big do we make these?
  - One way would be to run K-means on this data
    - Creates three clusters, even though they are not really there
    - Look at first population of people
      - Try and design a small T-shirt which fits the 1st population
      - And so on for the other two
    - This is an example of market segmentation
      - Build products which suit the needs of your subpopulations

Outline

- Introduction to Unsupervised learning
- K-means algorithm
- Optimization Objective
- Random initialization
- Choosing the number of clusters

K-means Optimization Objective

- Optimization objective

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K) = \frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)} - \mu_{c^{(i)}}\right\|^2$$

  - Called distortion (or distortion cost function)

$$\min_{\substack{c^{(1)}, \ldots, c^{(m)} \\ \mu_1, \ldots, \mu_K}} J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$$

  - $c^{(i)}$
    - Index of cluster $(1, 2, \ldots, K)$ to which $x^{(i)}$ is currently assigned
  - μ_k
    - cluster centroid k ($\mu_k \in R^n$)
  - $\mu_{c^{(i)}}$
    - cluster centroid of cluster to which $x^{(i)}$ has been assigned

K-means Algorithm

- Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in R^n$

Repeat {

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

for $k = 1$ to K

μ_k := average (mean) of points assigned to cluster k

}

K-means Algorithm

- Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in R^n$

Repeat {

Cluster assignment step

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

Minimize $J(\ldots)$ wrt $c^{(1)}, c^{(2)}, \ldots, c^{(m)}$
(holding $\mu_1, \mu_2, \ldots, \mu_K$ fixed)

Move centroid

for $k = 1$ to K

μ_k := average (mean) of points assigned to cluster k

}

Minimize $J(\ldots)$ wrt $\mu_1, \mu_2, \ldots, \mu_K$

Outline

- Introduction to Unsupervised learning
- K-means algorithm
- Optimization Objective
- Random initialization
- Choosing the number of clusters

K-means Algorithm

- Randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in R^n$

Repeat {

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

for $k = 1$ to K

μ_k := average (mean) of points assigned to cluster k

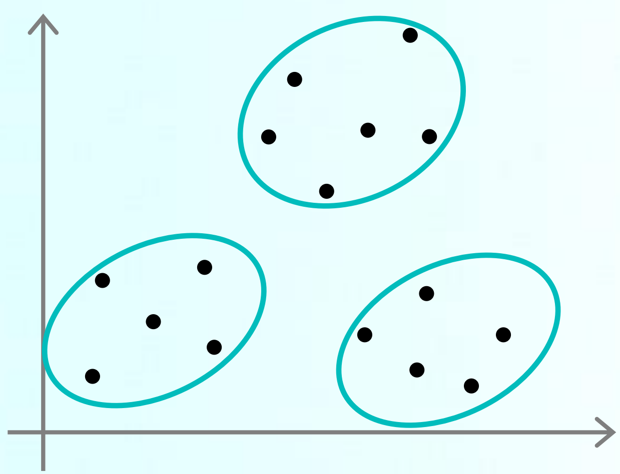
}

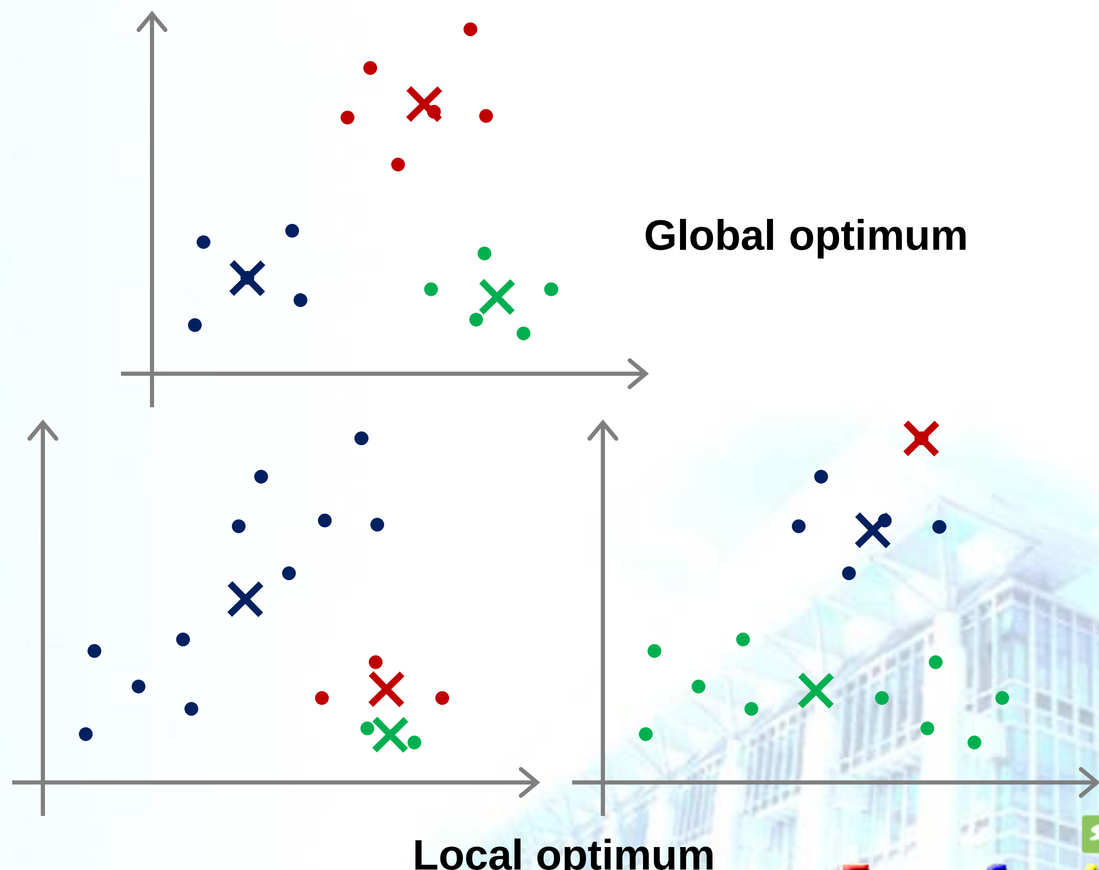
Random Initialization

- Should have $K < m$
- Randomly pick K training examples.
- Set $\mu_1, \mu_2, \ldots, \mu_K$ equal to these K examples.

Local Optima

- K means can converge to different solutions depending on the initialization setup
  - Risk of local optimum
  - The local optimum are valid convergence, but local optimum not global ones





Random Initialization

- K means can converge to different solutions depending on the initialization setup
  - Risk of local optimum
  - The local optimum are valid convergence, but local optimum not global ones

- If we concern a local optimum,
  - we can do multiple random initializations
    - See if we get the same result
      - Many same results are likely to indicate a global optimum

Random Initialization

For $i = 1$ to 100 {

Randomly initialize K means.

Run K means.

Get $c^{(1)}, \dots, c^{(m)}, \mu_1, \mu_2, \dots, \mu_K$

Compute cost function (distortion)

$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$

}

Pick clustering that gave lowest cost $J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$

Random Initialization

- A typical number of times to initialize K-means
  - 50 ~1,000

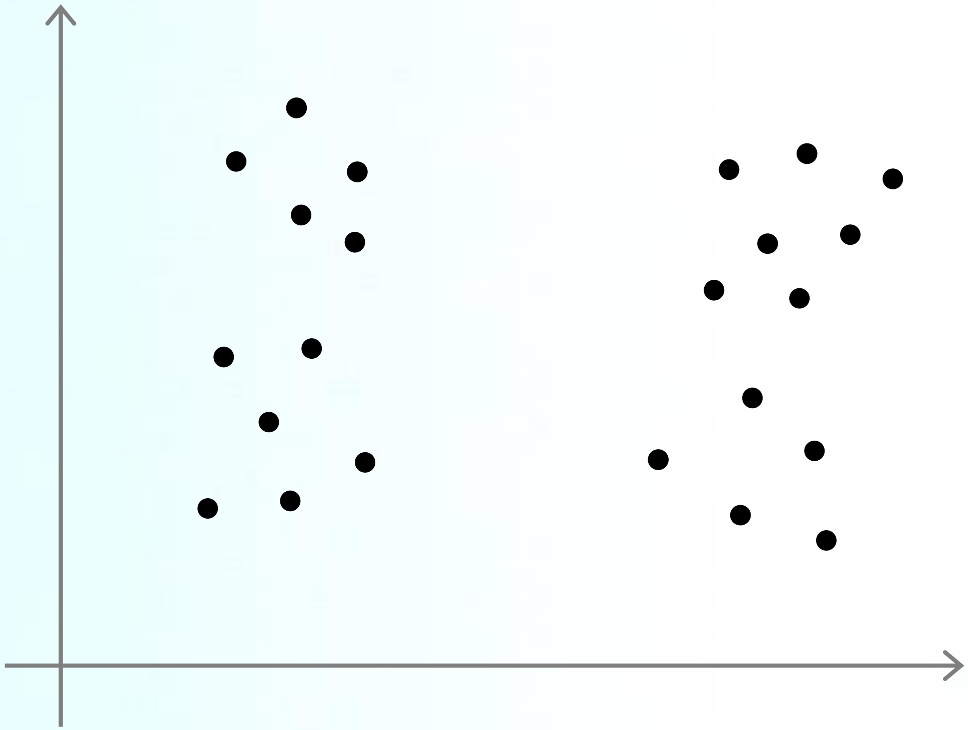
- If we are running K means with 2-10 clusters,
  it can help find better global optimum
  - If K is larger than 10, then multiple random initializations are less likely to be necessary
  - First solution is probably good enough (better granularity of clustering)

Outline

- Introduction to Unsupervised learning
- K-means algorithm
- Optimization Objective
- Random initialization
- Choosing the number of clusters

Choosing The Number of Clusters

- What is the right value of K?

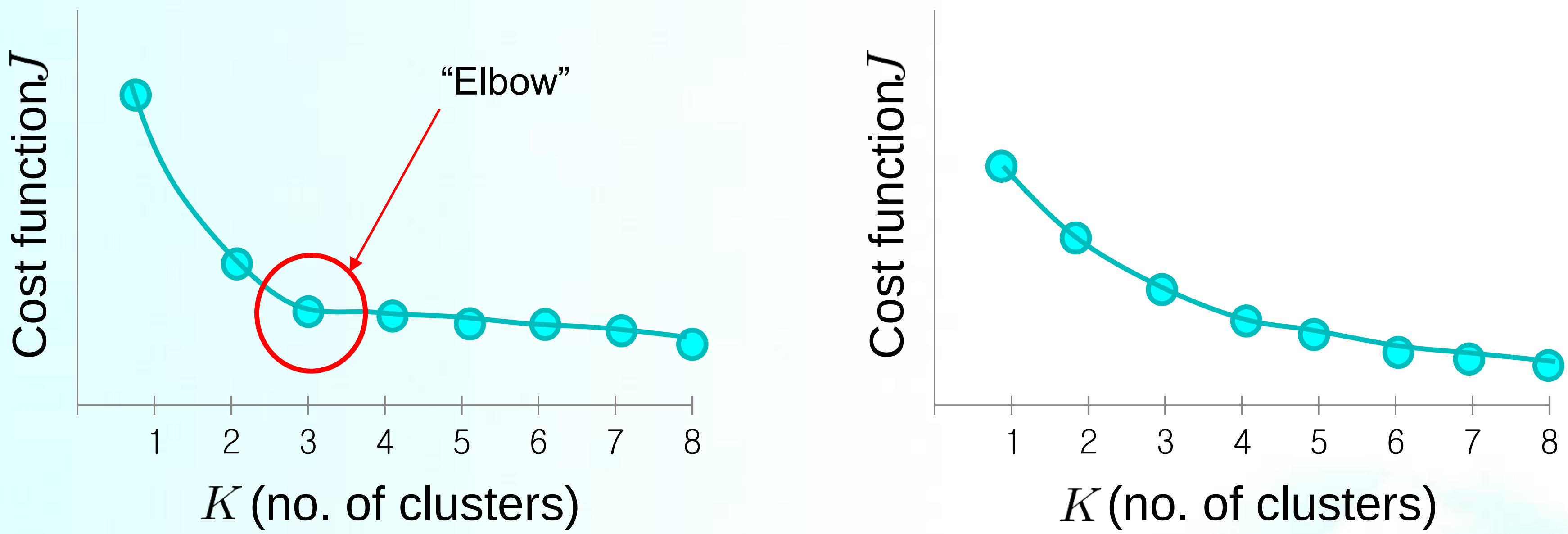


Choosing The Number of Clusters

- Choosing K?
  - Not a great way to do this automatically
  - Normally use visualizations to do it manually
- What are the intuitions regarding the data?
- Why is this hard
  - Sometimes very ambiguous
    - e.g. two clusters or four clusters
    - Not necessarily a correct answer
  - This is why doing it automatically this is hard

Choosing The Number of Clusters

- Elbow method
  - Chose the "elbow" number of clusters

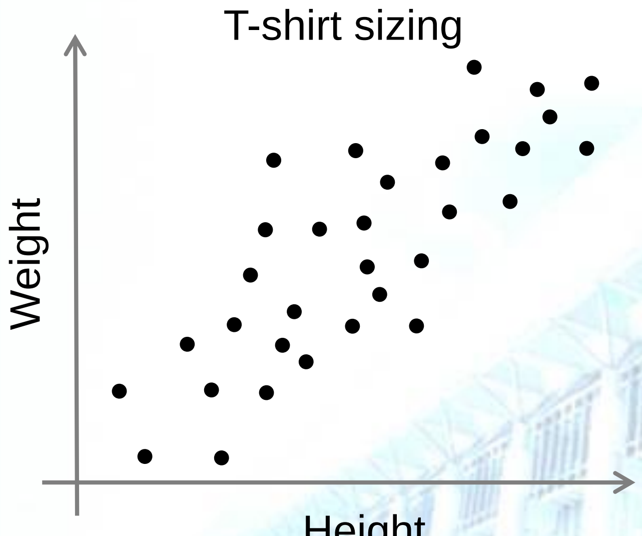


  - Risks
    - Normally, no clear elbow on curve
      - Not really that helpful

Another Method for Choosing K

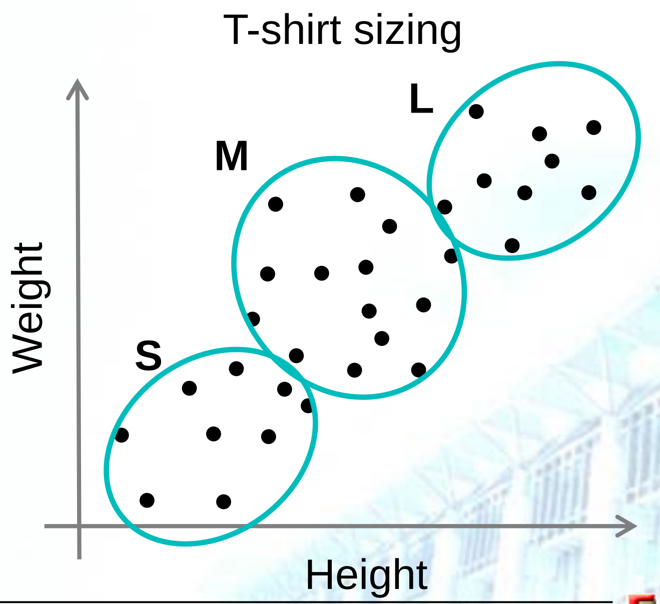
- Using K-means for market segmentation

- T-shirt size problem
  - Consider a company, which will release a new model of T-shirt to market.
    - In order to satisfy people of all sizes, models in different sizes need to be made.
      - So the company make a data of people's height and weight, and plot them on to a graph, as follows.



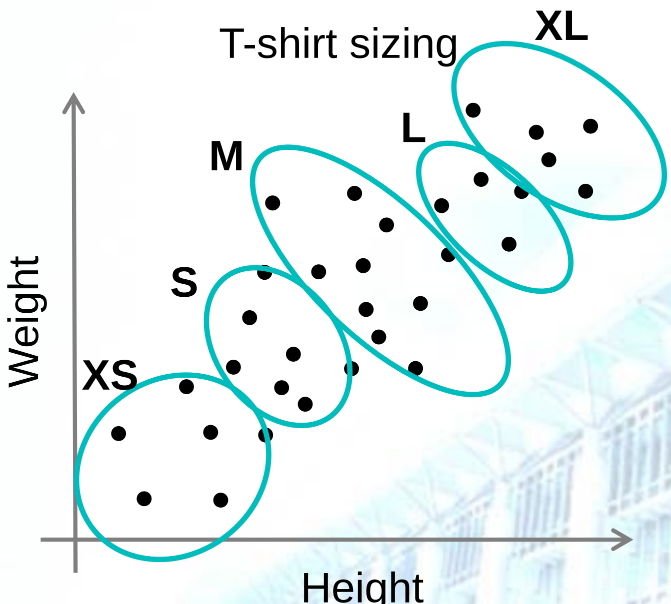
Another Method for Choosing K

- Company cannot create t-shirts with all the sizes.
  - Instead, they divide people to Small, Medium and Large, and manufacture only these 3 models which will fit into all the people.
    - This grouping of people into 3 groups can be done by k-means clustering, and algorithm provides us best 3 sizes, which will satisfy all the people.



Another Method for Choosing K

- Company cannot create t-shirts with all the sizes.
  - Instead, they divide people to Small, Medium and Large, and manufacture only these 3 models which will fit into all the people.
    - This grouping of people into 3 groups can be done by k-means clustering, and algorithm provides us best 3 sizes, which will satisfy all the people.
      - And if it does not, company can divide people to more groups, may be five, and so on.



Another Method for Choosing K

- Company cannot create t-shirts with all the sizes.
  - Instead, they divide people to Small, Medium and Large, and manufacture only these 3 models which will fit into all the people.
    - This grouping of people into 3 groups can be done by k-means clustering, and algorithm provides us best 3 sizes, which will satisfy all the people.
      - And if it does not, company can divide people to more groups, may be five, and so on.

➔ This gives a way to chose the number of clusters

- Could consider the cost of making extra sizes vs.
  how well distributed the products are
- How important are those sizes though?
  - e.g. more sizes might make the customers happier

References

- https://www.coursera.org/learn/machine-learning
- http://www.holehouse.org/mlclass/13_Clustering.html